Phishing on Open WLANs: Threat and Preventive Measure

Isha Khanna

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of
Master of Science
In
Electrical Engineering

Dr. Yaling Yang, Chair
Dr. Thomas Hou
Dr. Anil Vullikanti

12/08/09
Blacksburg, VA

Keywords: Phishing, Rogue AP, SSL, Certificate

Phishing on Open WLANs: Threat and Preventive Measure

Isha Khanna

ABSTRACT

Phishing is an internet security issue whose shape is still changing and size is still increasing. This thesis shows the possibility of a phishing attack on open, private Wireless LANs. Private WLANs which use a login page to authenticate users in hotels, airports and academic campuses are all vulnerable to this attack. Virginia Tech's WLAN is used as an example to show that the attack is possible. The attack combines two very well known attacks: one is to deceptively guide a user into logging into a fake website, which shows similar log-in page to the page of the website the user intends to go to, and the second attack is to show users a valid certificate, which does not show a warning. The rogue server takes the user to a log-in page which is similar to Virginia Tech's log-in page and shows him a valid security certificate.

We present a solution to the proposed problem. Software is implemented that runs on Windows Vista. The software warns the user if there are servers with more than one type of security certificates, claiming to be from the same network. We contrast our method to already existing methods, and show in what respects our solution is better. The biggest advantage of this method is that it involves no change on the server side. It is not necessary for the users to have any prior knowledge of the network, which is very helpful when the users access WLAN at airports and hotels. Also, when using this method, the user does not need to connect to any network, and is still able to get a warning. It however, requires the user to be able to differentiate between the real and fake networks after the user has been warned.

DEDICATION

To Mom and Dad,
and
To Ajay,

For your constant love, support and encouragement

Acknowledgements

I want to thank Bhagwan Sri Sathya Sai Baba. This is Your work.

I would like to express gratitude to my parents, Dr. Ashok Khanna and Mrs. Anjali Khanna, who have always had great confidence in me. I want to thank Ajay, my husband, for continuously motivating me and being immensely supportive. Thanks for giving this work the highest priority. Your good wise words constantly acted as an energy booster. My dad deserves special thanks for looking at the content and organization of the thesis. I also want to thank my second set of parents for their love and prayers.

I would like to thank my advisor Dr. Yaling Yang for her guidance and support. I was delighted to work in her lab, with the other students. I would also like to thank my committee members Dr. Hou and Dr. Vullikanti for their valuable advice and suggestions for betterment of my work.

My friends Shubhangi and Umesh deserve special thanks for all their support. The graduate student life at Virginia Tech will be a cherished memory for me because of them. I also want to thank my friend Rohit Rangnekar, who helped me improve the content and layout. My brother Mr. Uday Mohan deserves credit for his ideas and helpful suggestions that I could incorporate in my thesis.

I would like to thank Mr. Steven Lee who answered my incessant emails and questions about Virginia Tech's WLAN. I am thankful to John Harris, and John Paul for their help with the set-up. I want to take this opportunity to thank my lab-mates Siyu, Yongxiang, Han, Jatin, and Yujun for all their help with the experiments. I would like to thank Ting, Zhenhua, Chuan, and Jingyao for helping me prepare for the presentation.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Problem Statement

The thesis questions the safety of open private WLANs in the wake of all known phishing attacks, and attempts to understand the ease in executing such an attack. The work studies a phishing attack to steal credentials of clients of a wireless network which use web-based authentication. The attack can be seen as a combination of two well known attacks. One attack is to spoof the access point of the real network, and the second is to spoof the webpage that it uses to authenticate the clients. The problem is elaborated in chapter 3. The objective of the thesis is twofold: to find out how easy it is to execute such an attack and to propose a solution that is at client level. We talk about an attack on Virginia Tech's WLAN as an example. The attack can be executed over any private, open WLAN. A private WLAN of a hotel, or a shopping mall, all would be under threat.

1.2 Motivation

How safe is Virginia Tech's Wireless LAN (VT_WLAN)? Can we fool the faculty of a reputed college like Tech and can we hack their passwords while they innocently log into their very trusted WLAN? Unfortunately, the answer is in affirmative. With advancements in hardware and software, carrying out a phishing attack has become much easier than it was in past. Private WLANs are susceptible to attacks because of open medium, insufficient software implementations and improper hardware configurations. Among all prevalent attacks, a rogue AP attack is one of the most dangerous types. A rogue AP is present on about 20% of all enterprise networks. The deployment, discovery, and compromise of APs have become much easier for attackers. A lot of researchers so far have based their solutions on the wireless traffic that the RAP generates. Another line of research has been on the changes the server can incorporate. The related research work is covered in chapter 3. A need for a user end solution is emphasized.

1.3 Organization of the Thesis

The work is in three parts: in the first part, it is shown how to launch the attack, in the second, the existing solutions are analyzed and a solution is proposed, and in the third, experiments are carried out to show the chances of logging into a fake network depending on the signal strength of and distance from real access points. The second chapter discusses the existing solutions to the problem, the third chapter briefly describes the SSL protocol, the fourth chapter presents the proposed solution, the fifth chapter summaries the experimental procedures and results, and the sixth chapter concludes the work with recommended areas for future work.

1.4 Intended Contributions

The thesis studies RAP phishing attack on open WLANs, taking Virginia Tech's WLAN as an example. Open networks that authenticate users at the application layer by using a secured login page are vulnerable to attacks where the rogue network presents the clients a similar page to the one they intend to look at. In a big enterprise, deployment of rogue APs such that it escapes the network administrator’s eyes is easy. As such, it becomes challenging to find such rogue APs in a facility. Most of the existing solutions are for network administrators. A user level solution to the problem is proposed, and implemented for a Windows Vista client. The feasibility of the proposed software is checked by doing real world experiments, and the performance is evaluated.

Chapter 2: Background

The last chapter discussed the motivation, problem statement, and the intended contributions. This chapter focuses on the related work on rogue access points, and other possible preventive measures that can be applied on the server side.

2.1 Definition of Key Terms

2.1.1 Phishing

It is the illegal and criminal process of hijacking a client to a fake site to steal sensitive information such as user-names, passwords and credit card details. The Anti Phishing Working Group defines phishing as "a criminal mechanism employing both *social engineering* and *technical subterfuge* to steal consumers' personal identity data and financial account credentials."

2.1.2 Rogue Access Points

A rogue access point is any Wi-Fi access point connected to a network that has been installed without authorization from network administrators, or has been planted by a malicious user to carry out phishing or a man-in-the-middle attack. Rogue Access Points (R.A.P.s) constitute arguably a big security threat to Wireless (Wi-Fi) networks today. It is not under the management of network administrators and does not necessarily conform to network security policies. In [4], the authors classify RAPs into 4 types: improperly configured, unauthorized, phishing and compromised. Our focus will be on phishing RAPs.

2.1.3 Access Point Spoofing

It refers to impersonation of an actual AP by fake access points by using its SSID (network name), and even BSSID (MAC address of AP).

2.2 The Phishing Epidemic

In the first half of 2009, APWG reported 37,165 of unique phishing reports in May, around 7% higher than last year's high of 34,758 in October. The number of unique phishing websites detected in June rose to 49,084, the highest recorded since April, 2007's record of 55,643. Brand

domain pairs increased to a record 21,085 in June, up 92% from the beginning of 2009. Banking trojan/password-stealing crimeware infections detected increased by more than 186% between Q4, 2008 and Q2, 2009. The total number of infected computers rose more than 66 percent between Q4 2008 and the end of the half, 2009 to 11,937,944, representing more than 54 percent of the total sample of scanned computers [33].

The latest phishing scam that was reported was on Gmail, Yahoo and AOL Email this October. 30,000 email log-in credentials from these websites were hacked and posted online. It was later removed, but some of the compromised emails had already sent spams. Users received an email from Yahoo telling them that Yahoo needed to verify their email account as it had been idle for a long time. In May 2009 there were reports of phishing attacks on Face book [6, 9, 15, 21, 30, 32]. Users got message from their friends that included a link to a webpage that appeared exactly like the Facebook login page, but was a fake one and stole their credentials and sent a copy of the same message to all their friends [15]. IRS published several warnings to tax payers this year on fraudulent use of IRS name or logo to gain consumers' financial details. Scams were reported where users got e-mails that claimed to be coming from the IRS. The mails lured the customers into giving all their credentials by telling them that they were due a tax refund [21]. Early this year in January, there were reports of phishing scam on Twitter. It issued a warning to users asking them to carefully look at the URL before signing in [32]. Bank of America and Citibank have also faced phishing attacks in the past.

2.3 Similar Attacks

As said before, the attack can be seen as a combination of two well known attacks. The AP spoofing attack which shows a fake login page and makes users login to a fake network is not new [2]. NetCraft's article [31] mentions how even the users looking for a "lock" can be fooled. The attack involves a real certificate for the fake website. In some phishing attacks, the attackers paste a picture of the lock in the URL. The users who look for a lock, but almost never see the certificate can get easily trapped to believe they are logging into the real site. The client browser will also not get a warning since the certificate would be legal. The warning published recommends that the users should check the SSL certificate before providing credentials. In [17], researchers try to implement a similar attack by two different ways.

2.4 Types of Phishing Attacks

Phishing has been classified into web-based phishing and exploit-based phishing [14]. All the attacks which exploit well-known vulnerabilities in popular web browsers to install malware on the victim's machine come under exploit-based phishing. This thesis does not focus on such attacks. Web-based phishing is the oldest kind of phishing which started in mid-90's and has improved with time. The attacks are mainly email-based, where victims are sent spoofed emails with links to fake websites. The innocent users login to sites which seem to be that of the legitimate organizations. Such attacks are successful because the phishers exploit the lack of knowledge of users about authentic and secured sites. They use visually deceptive text in URLs which users do not pay attention to. Visually deceptive phishing can be further classified into using deceptive text, or copying images in the URL [12]. Users can be fooled by making fake websites with a very minor difference in the spelling of the domain name. For example, a difference between the letters "l" and "i" can escape the untrained eye, and "paypal.com", and "paypai.com" can appear to be the same. Phishers also use non-printing characters and non-ASCII Unicode characters [22]. Copying images of hyperlinks and copying images of the "lock" which indicates secured sites is also not uncommon. This thesis attempts to launch an attack which exploits such vulnerabilities of users.

2.5 Existing solutions

The problem that is being addressed in this work is an attack that concerns two areas, spoofing the website and the access point, and thus, the solution should also come from both the areas. This section discusses the existing solutions for finding rogue APs and also about the anti phishing tools to act as a shield against phishing websites.

2.5.1 Detecting Rogue A.P.s

Research on phishing attacks due to RAPs mostly follows monitoring the traffic and using that information to differentiate between the real and fake A.P.s.

2.5.1.1 Traffic Analysis

In [4], the authors propose an automated solution which can be installed on any router at the edge of the network. They first determine if the traffic is Ethernet traffic of WLAN traffic by analyzing inter-departure time of packets on a gateway. The next step is to track the attacker by analyzing the traffic he sends. This is based on the fact that the malicious user when gains access to a WLAN host connected to a RAP will perform port scanning to find out all vulnerable hosts. That traffic analysis is the basis of finding RAPs. However, such solutions can be used only by the network administrator. Also, because it depends on analyzing actual, real time traffic, these methods are impractical to efficiently apply in big networks.

2.5.1.2 Radio Infrastructure

In [25], a framework for network fault diagnosis and security is provided for by using a dense deployment of sensors. Desktop machines need to be deployed all over the facility, with a USB wireless card, and two softwares installed. One is device driver software, and another is a user-level code. It listens in both active and passive modes, in a promiscuous mode. They modify the device driver to copy all the received 802.11 frames in the kernel buffer. With each frame, details like the signal strength, the channel and the data rate are also stored. This information is made available to the users. The service has a filter capability, which analyzes frames and makes the information available to the user. The method does suggest tests to reduce false alarms. In totality, the solution is not very easy to implement, requires more hardware, and may raise false alarms [17]. This approach is similar to Cisco's commercial product: Wireless LAN Solution Engine[8]. One big limitation of this method is that if the 802.11 feature of sending out beacon frames is switched off, the method will fail to work [4].

2.5.1.3 Sniffers

Another method is to use "sniffers". Sniffer softwares let the users carry a laptop or PDA around the wireless network area scanning all radio frequency (RF) channels for connections with all the access points within range. Ex: AirMagnet or NetStumber. The process is called "wardriving". It can be very time consuming to walk through all facilities in search of rogues. This also can be used by network administrators only, since a user can identify real APs only if he has a prior

knowledge of IP or MAC addresses of actual access points. If the phishers are aware of an ongoing scan by a network administrator, they can unplug the rogue APs at that time, or can program the AP such that it doesn't send out beacon frames.

2.5.1.4 Probes

They are small hardware devices designed to collect all wireless data passively [3]. To ensure continuous vigilance for rogue APs, full-time probes can be installed. Installing probes can be an expensive proposition. Because they operate continuously, dedicated probes are more reliable than using sniffers where someone has to walk around with a laptop. While dedicated probes allow network administrators to aggregate information about all APs operating within the range of your wired network, the network administrator still needs to determine which of those foreign access points are actually rogue. This also takes a lot of time. The drawbacks are the same as that of sniffers.

2.5.2 Application Layer Level Solutions

2.5.2.1: Heuristic Based Phishing Methods

These schemes try to identify phishing websites by examining the URLs of the websites. Many browser plug-ins are available which warn the users using this technique. Spoofguard, CallingID, are such plug-ins [26], [7]. Spoofguard, for example, warns users against phishing websites based on a number of factors. It uses domain name, URL and image checks to identify a suspicious link. Spoofguard uses history, which means it makes checks like whether the website has been visited before and whether the referring page was from an email. It also examines user name and password fields in post data. A spoof index is defined depending on the above mentioned variables, and their weights, and the total spoof index of a page is used to determine the severity of the alert. These toolbars would work in most of the cases against a phishing website, but only after the user has got access to internet. Some other, more complex ways involve measuring web page visual similarity [41]. Web pages are converted into low resolution images and then color and co-ordinates of those images are used to make image signatures.

2.5.2.2 Restriction Lists

The most common method utilized by many anti-phishing tools is the use of whitelists and blacklists. Blacklists are made after phishing reports are received [26]. Some examples of such tools are Mozilla's Web of Trust, and Ebay's toolbar. Web of Trust [40] warns the users against websites that might pose a phishing threat if the sites have been known to send spam or deliver malware. The color coded icons of WOT indicate whether the site can be trusted or not. Ebay toolbar is a similar tool. The drawback is that new spoof sites will be reported as phishing sites only after some time period, during which users will be fooled [13]. Such solutions are not feasible for a rogue AP attack, since the user will have to login to one of the networks before he can use any of these tools.

2.5.3: Other Solutions

2.5.3.1 Multi-factor User Authentication

The server uses a combination of "something that the user is", and "something that a user has" to authenticate him. Ex: AOL Passcode, CRYPTOCard. AOL and RSA Security launched AOL Passcode, a device that generates and displays a unique six-digit numeric password every 60 seconds [13]. It's a one-time password that is used along with the usual password to authenticate the user. CRYPTOCard is a company that has a technology to issue similar tokens to companies [10]. If the secure site uses a Cryptocard, at the server side, the user at-least is making sure his username/password will not be used again to create mischief [11]. It's not very costly to use. It does require the user to carry a device with him. It mandates a change on the server side.

2.5.3.2: Using secured connection

For WLANs like that of Virginia Tech's, if the user installs the certificate beforehand he can log into the wireless network without the threat of attackers stealing his password. At VT, the SSID is seen as "VT-Wireless". Hotel or airport WLANs cannot distribute their certificates in advance, so this is not a generic solution.

Application layer level solutions cannot be used in the mentioned attack, because these solutions work only after the client has acquired internet access. As such, the focus of this work will be on rogue AP detection schemes.

This chapter focused on the research that has been done towards finding out rogue APs. It also suggests a few preventive measures that can thwart such attacks to some extent. The next chapter discusses the SSL protocol, and how the client and server secured interaction takes place. It then describes the problem, and explains why the attack is possible to execute.

Chapter 3: SSL Security

This chapter goes over some basic concepts of public key cryptography. It then describes the problem in detail and answers questions like how the attack was launched and why it was possible to do so.

3.1: Public Key Cryptography

Communication on the internet involves many hops before the data sent by the client can reach the server. The data is susceptible to attacks like eavesdropping, tampering, and impersonation. These issues are addressed by public key cryptography. It facilitates encryption, authentication and non-repudiation. For this purpose, public key encryption involves a pair of keys.

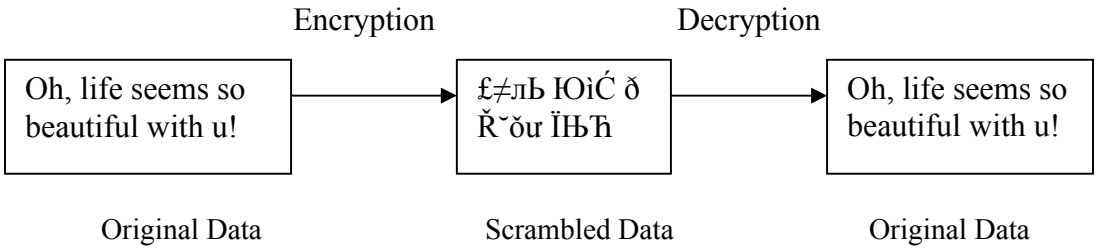

Figure 3.1: Public Key Exchange

Figure 3.1 shows a simplified view of the way public-key encryption works [19]. Each party creates its own pair of public and private keys. The public key is distributed, so that anyone sending him a message might encrypt it by using his public key. The public-private key pair works such that a message encrypted with one can be decrypted using another. If anyone knows the public key, and the algorithm, he can still not compute the private key. Data encrypted with the public key can be decrypted only with its private key.

Let us take the example of Alice and Bob. If Alice needs to send a message to Bob, she will encrypt it using Bob's public key. Bob can decrypt it using his private key. The one problem that still remains is that the public key has to be distributed in a trustworthy fashion. Bob wants to be sure that he has Alice's public key, because there can be a Eve, who might send public keys

saying she is Alice. This problem is solved by PKI. Figure 3.2 shows how public key cryptography works in this example.

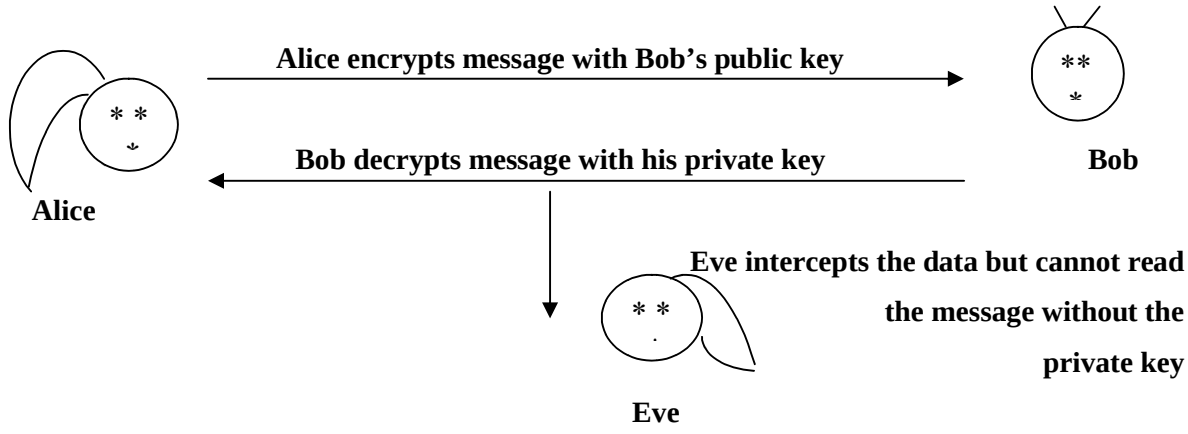

Figure 3.2: Public Key Cryptography steps showing Alice sending Bob an encrypted message using his public key, Bob decrypts using his private key, Eve is shown eavesdropping

The SSL protocol uses PKI to secure the communication between a client browser and a server. The protocol was created by Netscape. It runs above TCP/IP and below the application layer protocols, as shown in figure 3.3. It allows an SSL-enabled server to authenticate itself to the client, allows the client to authenticate itself to the server, and lets them both establish an encrypted connection [20].

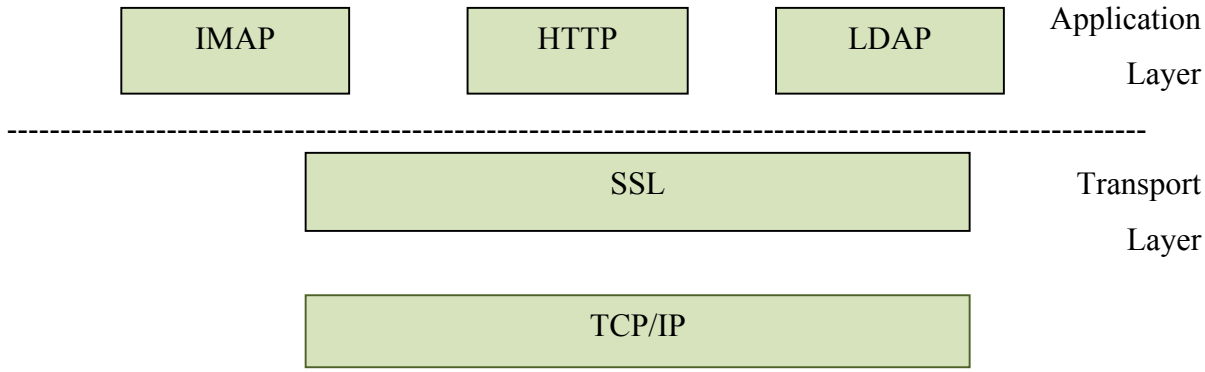

Figure 3.3: SSL in Protocol Stack

PKI is an infrastructure that ties together the public keys with respective user identities. This is done by getting a certificate, which acts like an identification, from a certificate authority (CA). CAs are trusted authorities, like Verisign, that issue a certificate to a user for his website, or

network. Users register on the site with a request to get a certificate, and the certificate is issued only when the user credentials have been validated. The PKI role that assures this binding is called the Registration Authority (RA). Each certificate issued by the CA contains the user's identity, the public key, it's binding, validity conditions and other fields like issuer name, encryption algorithm. Compared with symmetric-key encryption, public-key encryption requires more computation and is therefore not always appropriate for large amounts of data. However, it's possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This is the approach used by the SSL protocol [19]. The protocol has two parts: the record protocol and the handshake protocol. The former lays the definition of the format to be used to transmit data, and the latter involves using that to exchange messages between the server and the client when a connection is established. The functioning of SSL can be summarized as the following [20]:

1. The client sends its version number, cipher settings, random data to the server.
2. The server sends the client its corresponding details. It also sends the client its certificate, requests client's certificate if client authentication is needed.
3. The client authenticates the server.
4. The client creates the pre-master secret for the session, encrypts it with the server's public key, and sends that to the server.
5. If the server demands client authentication.
6. Both client and server generate the master key and session keys from the pre-master key. These are symmetric keys.
7. The client and server each send each other an encrypted message.

3.2: SSL Certificates

Users of a public key require confidence that the associated private key is owned by the correct remote subject (person or system). This confidence is obtained through the use of public key certificates, which are data structures that bind public key values to subjects. The binding is asserted by having a trusted CA digitally sign each certificate [18]. Digital certificates are issued by a trusted third party known as a Certification Authority such as VeriSign or Thawte. These third party certificate authorities confirm the identity of the party demanding a certificate, and

provide assurance to the website visitors that the website is one that is trustworthy [38]. The certificate issued by the CA binds a particular public key to the name of the entity the certificate identifies, such as the name of an employee or a server. Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate will work with the corresponding private key possessed by the entity identified by the certificate [19]. The certificates encrypt the public key of the website/user by their own private key. When the client's web browser attempts to secure a connection, the digital certificate issued for that website is checked by the web browser to be sure that all is well. The web browsers come with all the main certification authorities and their public keys inbuilt in them. They use that information to decrypt the digital signature. When a client visits the server's website, it is presented with the certificate. The client browser makes sure it has been signed/issued by a trusted third party. In IE, the list of trusted certification authorities can be seen by going to "Tools->Internet Options->Content->Certificates->Trusted Certification Authorities". Figure 3.4 shows a screenshot of this list. The browser checks if the certificate is valid, in time and if the certificate common name and the website name match. After that, the keys are exchanged for a secure transaction after this level. That follows the SSL protocol. This allows the browser to quickly check for problems, abnormalities, and if everything checks out the secure connection is enabled. When the browser finds an expired certificate or mismatched information, a dialog box will pop up with an alert [38].

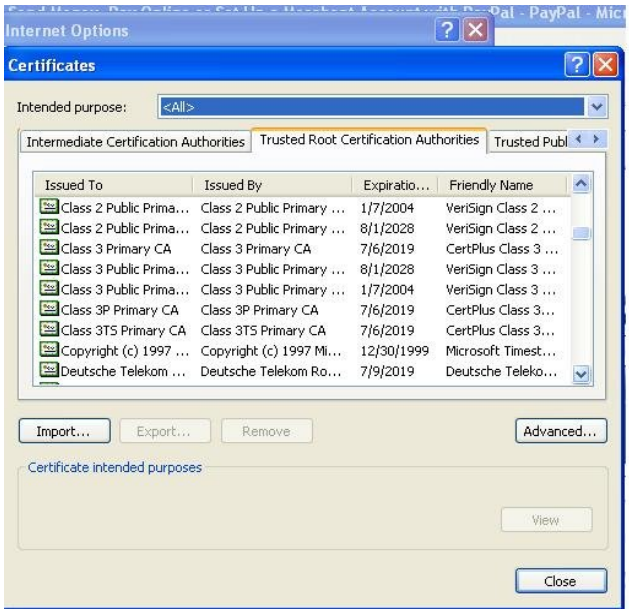

Figure 3.4: List of Trusted Certification Authorities

Digital certificates have two basic functions. The first is to certify that the website or the servers/routers are who or what they claim to be. The second function is to keep the data exchanged between the client and the server secure. It is extremely important for e-commerce, like for credit card transactions, where it protects against theft [38]. This is a snapshot of Paypal's security certificate as viewed from IE:

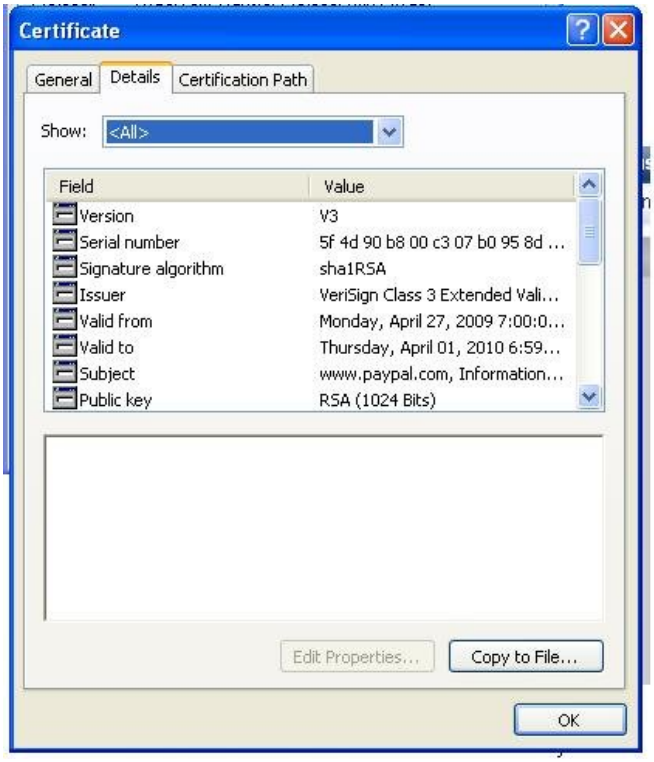

Figure 3.5: Certificate Contents

In general a certificate has the subject name, the subject's public key, the issuer name, the period of validity, signature, and other optional fields. Figure 3.5 shows the contents of a certificate as viewed from Internet Explorer. The signature the hash code of other fields, encrypted with the CA's private key. The browser has the public key of all the trusted CAs, and uses that to make sure the certificate has been signed by trusted authority.

The "Distinguished Name" field is looked at in detail. This is the field that is used later to differentiate two websites with similar login page. An X.509 v3 certificate binds a distinguished name (DN) to a public key. A DN is a series of name-value pairs that uniquely identify an entity-

-that is, the certificate subject. For example, this might be a typical DN for an employee of Netscape Communications Corporation: UID=doe, E=doe@netscape.com, CN=John Doe, O=Netscape Communications Corp., C=US, where UID is the user ID, E is email address, CN is the user's common name, O is for organization and C is country. The “SubjectName” field in the above snapshot is “paypal.com”, confirming that this certificate belongs to Paypal.

3.3 Breaches in current SSL system

SSL is susceptible to man-in-the-middle attacks [28], [29]. The rogue server can act as a proxy server, and get in between the client and the actual server when the key exchange is going on. I can give its own, fake, key and certificate to the client. This is a layer 2 attack which can be executed if the proxy server does DNS spoofing, and sends its own IP address to any DNS query that the client server generates [28]. Another attack on SSL is the MD5 attack. MD5 collision attack can be used to forge certificates. The researchers have managed to create two different messages with the same digital signature using MD5 [29].

3.4 Problem Description

3.4.1 Why is the attack possible?

In [12], the authors find out the reasons that make phishing possible. The attackers exploit the weaknesses of clients. The attacks can be based on lack of knowledge of users. Users may not be able to differentiate between a secured site and an unsecured site. A lot of people now know the meaning of security icons. Even if they know what signs they have to look for, very few people understand CA certificates. A common user never checks the certificate, in fact, 95% users do not even pay attention to the certificate warning, and go ahead even if certificates are not from trusted authorities. Another vulnerable area is the website URL. Phishers use deceptive text to fool users into logging into fake websites which have URLs very similar to the URL of the website they want to go to.

3.4.2 The attack

The attack addressed in this thesis is a combination of AP spoofing and website spoofing. The attack can be launched on any secure private WLAN if the WLAN uses a login page to first verify users before giving them access to internet. Figure 3.6 shows the attack setting.

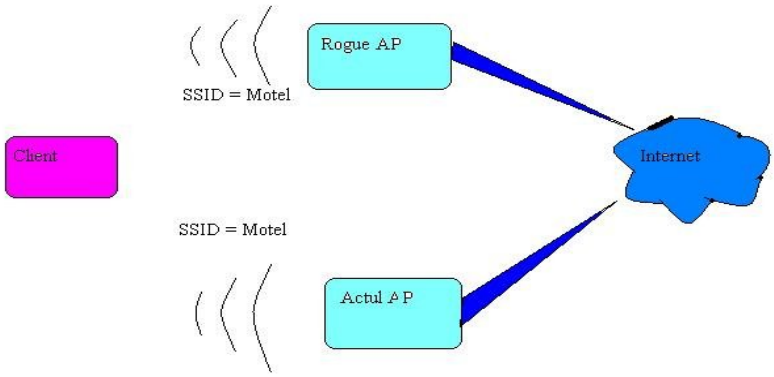

Figure 3.6: RAP Attack Setting

WLANs like that of a hotel, or airport, or Virginia Tech campus present the clients a login page before giving them access to internet. The client connects to the Wi-Fi network, enters his username and password and when authenticated, gains access to internet. If the attackers deploy a rogue AP in the same area with the SSID same as that of the real network, the client can log into the rogue network. If next, the attackers present to him a page that is similar to the login page of the real network, he can be fooled to login to the fake network. If the real network uses SSL, and if the user is aware of that, he may look for an "https" in the URL, and for other security icons like the lock. To fool such users, the phishers can get a certificate corresponding to the fake website. For example, if the hotel website is https://www.hotel1.com, the phishers can name their webpage https://www.hotell.com and get a certificate issued from a CA authority for this fake website. Once the user logs into the network, the rogue network can use his id and password to log into the real network, and give the user internet access by forwarding traffic to the real network. Unless the user is so suspicious that he checks the security certificate, reads and matches the subject name and the website URL, he will be fooled to log into the fake network.

The next chapter underlines the challenges involved and the details of the proposed methodology.

Chapter 4: Proposed Methodology

This chapter highlights the challenges of the work, in the light of drawbacks of the available methods. The design of the proposed solution is explained with the help of a flowchart and algorithm.

4.1 Challenges

The research for detecting rogue access points (APs) is mainly based on traffic analysis, as described in the previous chapter. In [13], the authors talk about a similar attack as discussed in this work, and according to them, the solution would be if a user himself compared certificates. Although, expecting users to know well enough about certificates to compare two of them is not realistic. There is a need for an automated software that detects rogue APs. To summarize, the solution should overcome the drawbacks of the existing methods and meet the following requirements:

- The implementation should be on the client side rather than on the server machine as the server changes are not accepted widely.
- It should not require any prior knowledge of the network the user is entering. It should work even when the user logs into the network for the first time.
- It should be time efficient.
- The client should not need to log into any network. Most anti-phishing tools are developed as browser plug-ins and work only after the user gets internet access.
- It should be cost effective.

4.2 Design Overview

The proposed system presented in this thesis aims at providing a client-level solution to the problem of possible phishing-rogue-AP attacks while logging onto a private open network. The software that is used as the solution is a combination of a Windows utility, and a code written in C#. The software works for any client machine with Windows Vista operating system. The software attempts to warn the user of any possible rogue APs in the vicinity by examining the distinguished names. The clients are recommended to run this software before actually connecting to any network.

Although the software does warn the users of existing rogue APs, it cannot differentiate the rogue network from the actual network. An assumption is that once a warning is issued, the user will be very careful while logging into any network, and will not enter credentials unless he has examined the URL of the login page, and is convinced of its authenticity.

The software first establishes a connection with the network the user selects, and then starts an internet browser. It programmatically extracts the security certificate from the login page of the network. This process is repeated a couple of times, selecting a different AP each time, so that the software logs into all the APs present near that point. It then compares all the certificates, and if there is a difference in the subject names associated, it alerts the user. The proposition is that the software can discriminate between the rogue and the actual network by examining the security certificates offered by both of them. The RFC 3280 [18] describes the fields of the certificates such as "Distinguished Name", comparing which the client can make an intelligent decision. It is evident that the "Distinguished Name" field will be different for the fake and real servers, because the URLs of the two webpages are different. When a CA issues a certificate to any server, it verifies the email id, name, and the organization [37]. The attacker's certificate will have a domain name different from the real network's domain name. A comparison between the two certificates will alert the user, and he should then be able to avoid logging into the fake network.

4.3 Attack Set-Up Procedure

The attack is aimed at private networks, where internet access is allowed after a secure login to a server. There is usually a login/welcome page, where the user enters his login information, and is then allowed internet access. We show how an attack is possible in such cases, and we take Virginia Tech's WLAN only as an example to show this. A similar attack can be done on any private, open network which uses SSL certificate. This section gives the details of how the attack was launched. Figure 4.1 shows the logical diagram of the attack.

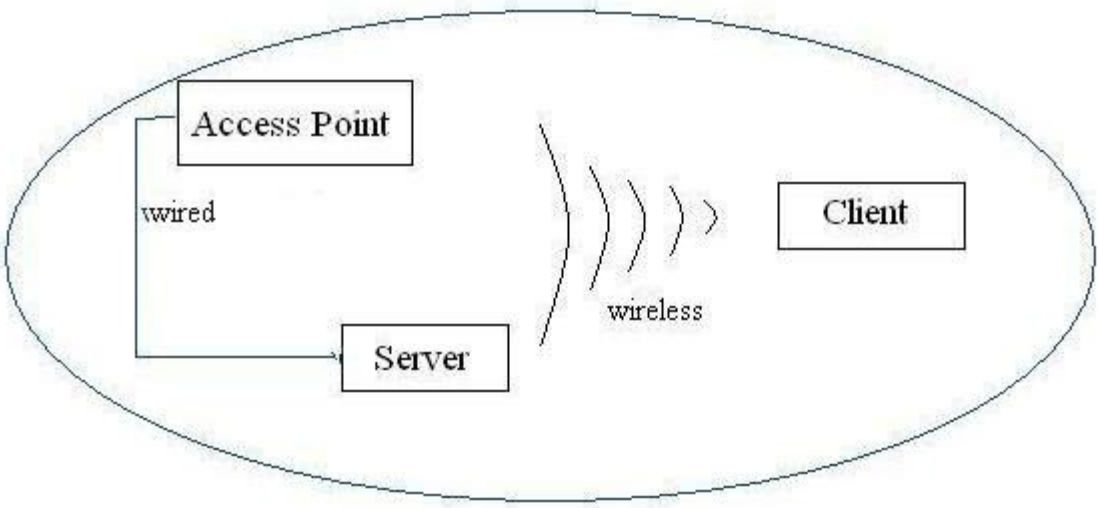

Figure 4.1: Logical Diagram of Attack

Intel's access point was used to act as a gateway and AP. A Linux machine was used as a server. Apache server was set up on the same machine to present the fake login page to clients. This research did not use an authentic certificate for our experiment. OpenSSL was used to create a self-signed root certificate (Appendix B). The following steps were followed to launch the attack:

1. Access point was made to work in the "Gateway + AP" mode.
2. SSID of the RAP was set to VT_WLAN.
3. The DHCP server was disabled on the AP since DHCP and DNS servers were set up on the server machine.
4. The server was connected to the AP via a LAN cable.
5. NAT router settings were done on the server [36].
6. DHCP [16, 33] and DNS [5, 34] servers were set up, and started.
7. Apache server was configured on the server. VT_WLAN's login page was copied and made the default page, and apache server was started.
8. OpenSSL was used to get a self-signed root certificate, for the login page.
9. On the client browser, the certificate of the server was added as one of the "Trusted Root Authorities" [35].

Once the user enters his credentials on the fake network page, the attacker can use it to login to

the actual network, and can then forward all user traffic to the actual WLAN.

4.4 Block Diagram of Software

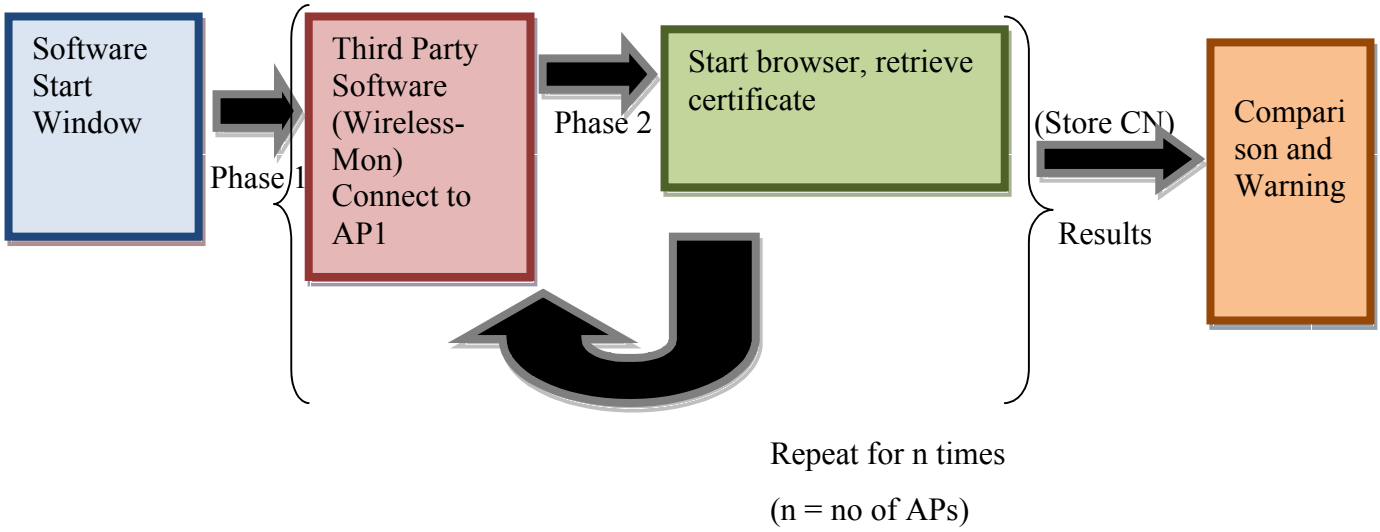

Figure 4.2: Block Diagram of Software

4.5 Algorithm

The code is written in C#, and can be used for computers using Windows Vista as the OS. The figure in section 4.4 represents the steps in a diagram. The algorithm for the software is thus:

1) Start the Windows software "WirelessMon" (Phase 1)
2) Connect to AP, looking at MAC address
3) Start IE, and send an http request, to google.com (Phase 2 starts)
4) Get the certificate from the site using C#s inbuilt utility
5) Store the subject name of the certificate in a file (Phase 2 ends)
6) Steps 2-5 are repeated for n times, where n=number of APs visible
7) Give result, i.e, warning if any difference in certificate domain names

This concludes the design of the solution proposed in this research work. The next chapter presents the results of carrying out experiments on VT_WLAN.

Chapter 5: Real World Experiments

5.1 Experimental Background

The chapter first shows how the client connects to the rogue server. The screenshot below shows the login page when the client sends a www.google.com request when connected to the fake network. The URL is https://www.google.com, where as if the client logs into VT_WLAN, even if the google.com request is sent, the URL is "https://www.webauth.cns.vt.edu". The login page was made similar to Virginia Tech's authentication page when the SSID was different, so that no innocent client saw this page. Figure 5.1 shows a screenshot of the fake webpage.

Figure 5.1: Fake Login Page

If the user pays more attention to the URL, is cautious, and looks at the security certificate assuming he knows about security certificates, he will find the certificate which should raise an alarm. The "Issued to" field has the subfields common-name (localhost 377lab), organization, etc. which are not coherent with Virginia Tech's details. Figure 5.2 shows these details.

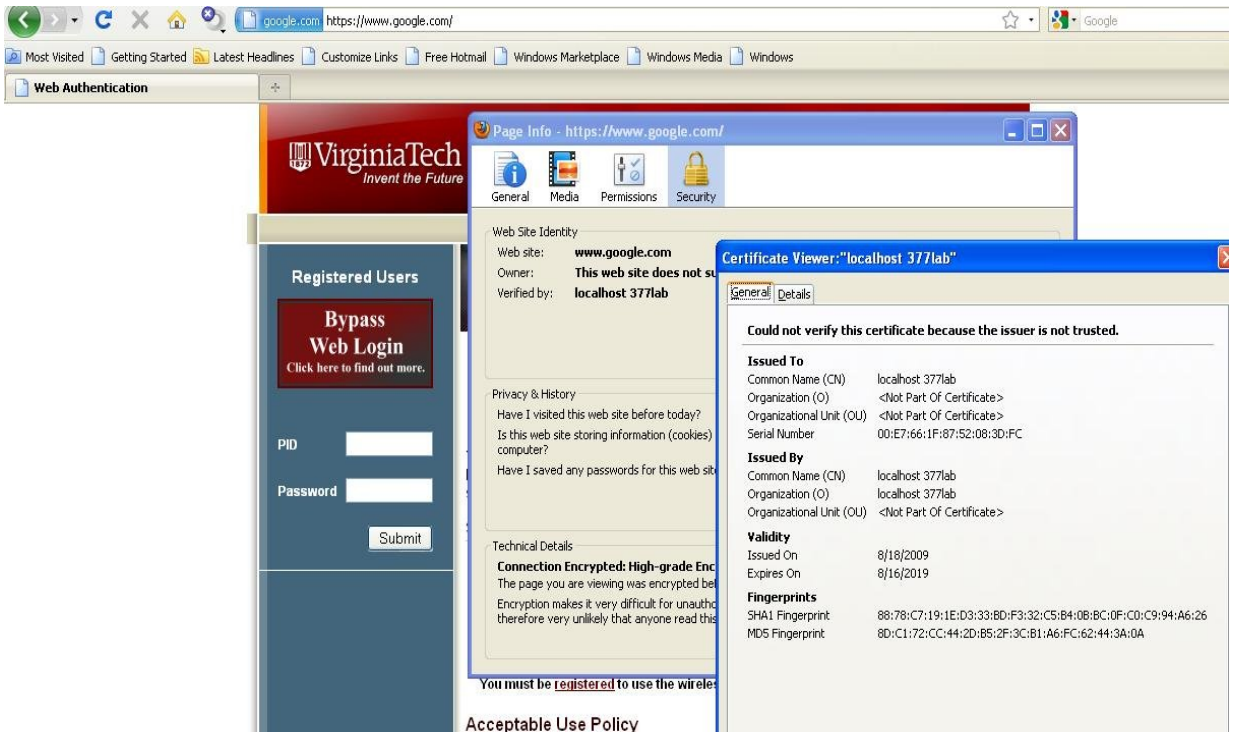

Figure 5.2: Fake Certificate

5.2 Logical Diagram

The RAP was kept between two actual APs, and the client was kept in several different positions between the three APs. The real APs are called AP1 and AP2. The distance between an actual AP and the RAP is "x", for example. The location of two APs was close enough to assume that the client would connect to only either of those two APs. The distance "x" in reality was about 7 feet. Figure 5.3 shows the logical diagram.

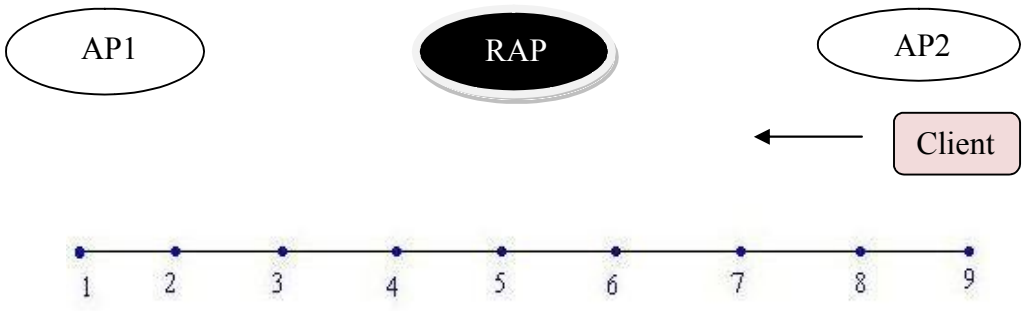

Figure 5.3: Experimental Set-up Block Diagram

5.3 Tests

The experiments were done in two parts. Both tests were done using the above set-up. The first one was done without equipping the client with the proposed software. In the second test, the client used the proposed software. First set of experiments were done to find the severity of such a threat. Second set of experiments were done to find out the feasibility of the proposed software. Rogue AP and the server were kept midway between two actual APs in one of the buildings at Virginia Tech. For experiments, the SSID was changed to VT_WLAN and the login page was changed to a disclaimer informing users of the ongoing research. The experiments were carried out after midnight, to make sure no wireless user suffered a denial of service. For security purposes, the MAC addresses and positions of the real APs are not revealed. The actual APs were Cisco APs, with a much stronger signal strength than that of the RAP, which was an Intel gateway. The client was moved along a line, with varying distance from actual and rogue APs.

5.3.1 Test 1: Client without software

The first sets of experiments were done without equipping the client with the proposed software. The aim was to find out the probability of a user getting connected to the RAP in presence of real APs and to highlight the severity of the problem. The client was kept at several positions between the three APs. It is assumed that the client is using a normal Windows utility to connect to a wireless network. He would then automatically connect to the AP with the strongest signal. (IBM laptops let the user connect to the AP of his choice). The client was made to connect to, and disconnect from a wireless network. The process was repeated ten times at each position. Figure 5.4 shows the signal strength of each AP at different points. Figure 5.5 shows the frequency at each point with which the client logged into the rogue network, an average was taken over multiple runs that were performed to get this graph.

As expected, the result was found to be depended upon the distance from, and the signal strength of the actual APs and the rogue AP. In the absence of the software, the user will log into the fake network with the probability p. This probability, will increase if the rogue AP is from a better company, like Cisco. These experiments were done not using any WLAN utility, and assuming that the client's machine will automatically login to the AP with maximum signal.

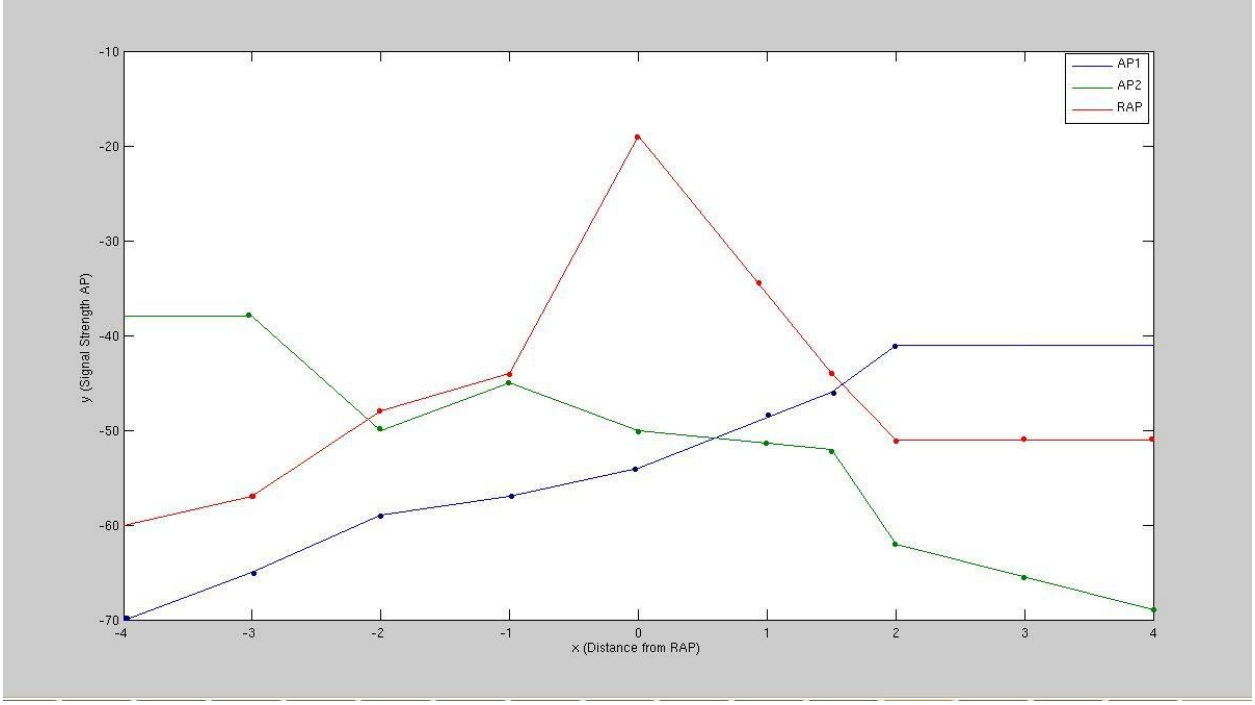

Figure 5.4: Signal Strength Graph

When the signal strength from any one AP is very high, -42db for example, and from other APs is less, with a difference of about 10, then the client always connects to the one with the higher signal (position 1, 2). When the strengths are comparable, then the probability of connecting to the RAP depends on the number of real APs around the client, and the distance from each AP (position 3). Based on the above experiments, a distance (with respect to the RAP) versus the probability (of connecting to it) graph can be plotted. The graph is shown below:

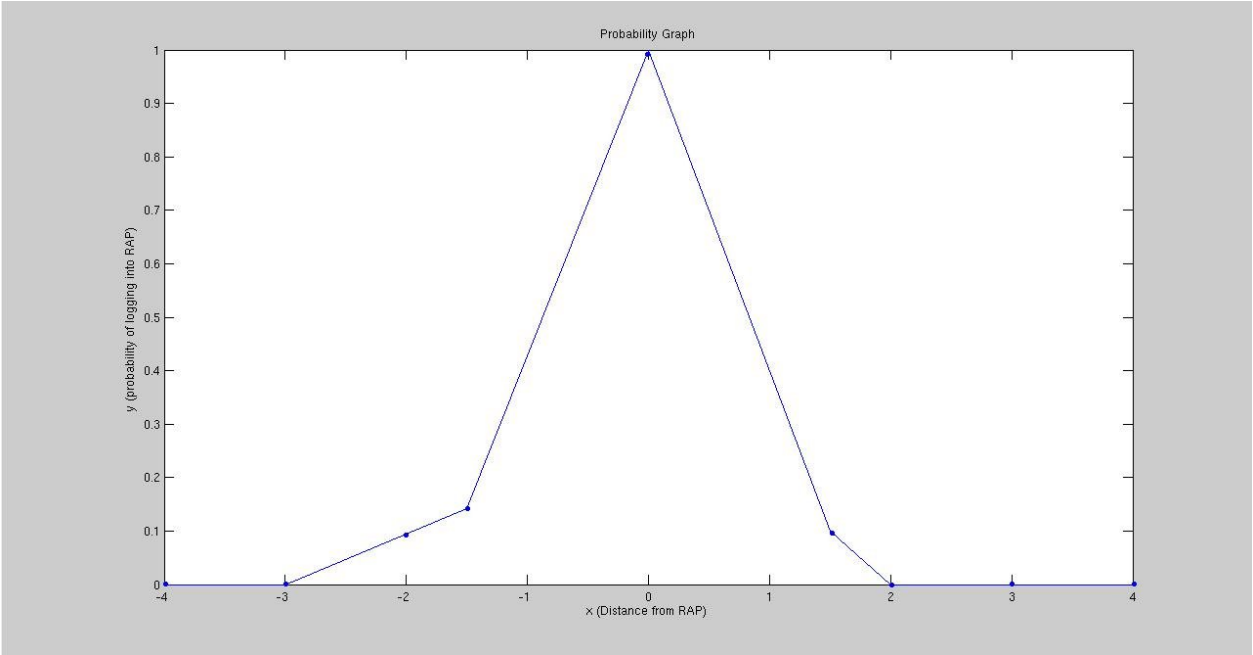

Figure 5.5: Probability of logging into RAP versus the distance from the RAP graph, averaged over multiple runs

The graph was found to be asymmetric, although it can be symmetric if plotted at a different location. Between points x=-3 and x=-1, and x=1 and x=2, the client logged into the fake AP with some probability, as shown in the graph. Between points x=-1 and x=1, the probability of the client logging into the fake AP is very high. Beyond x=-3, and x=2, the client connects to the actual APs. The reason is that the client is physically very close to the RAP. In this case, the probability of the client being at any of the nine points is the same.

There are 5 positions out of 9, where the client would log into the actual AP with probability 1. At the other 4 points, the probability of logging into the RAP is .1, .5, 1 and .4 respectively. VT uses Cisco APs and this work used an Intel gateway for the rogue network set up. Also, the probability of the client logging into the rogue network even when he is placed between the actual and the rogue AP is not .5 because the APs are from different manufacturers.

5.3.1 Test 2: Client equipped with the proposed software

The second sets of experiments were done when the client was using the software. The aim was to find out the performance of the proposed software, and contrast it with first set of tests. The set up was the same, and the client was put in different positions with respect to the three APs as

before. As against the previous tests, this time the software was used, so the client did not automatically get connected to the AP with the highest signal strength. In fact, it was made to login to the three APs present. Since the APs were visible from every position, the client could log into each AP from any of the nine positions pointed out. It was found that the client got a warning in every position. So the performance of the software does not depend upon the physical location of the client in this case.

5.4 Performance

The software was found to effectively warn the client of the presence of a RAP asserting that there were two different certificates from the same network. The measure of performance on the basis of time will be vague, as a third party software was used. The total delay will be the sum of delay in connecting to an AP, delay in pulling up the internet browser, and that in extracting and comparing the certificates. Out of these three, the first depends on the number of APs present, and the sum of second and third was found to be 9 seconds. This delay also, however, depends on the client's computer's efficiency. There will be a trade-off between the delay and the number of runs, which will be equal to the number of APs visible to the client. If programming is done at the driver level, such that the driver can start parallel threads to connect to APs, it would be possible to reduce the delay to only a few seconds. In the test case, the total delay in connecting to an AP, and storing it's certificate was found to be about 44 seconds.

One special case will be when the rogue network does not have a secured webpage. It is possible to copy the login page and not show an "https" URL. In that case, the client will be alerted about this as well. The fact that there are two different security levels of seemingly one network is also a cause for caution.

In rare cases, it is possible that two different networks with the same SSID exist in one locality. They both should then have different login pages. The user will be alerted, but when he logs in, he will not be shown the same page by the other WLAN. It will be a false alarm. The special cases can be further analyzed so that the user never gets a false alarm.

The software will fail to work in case the client cannot see two different types of APs. That will be the “corner case”, where the signal from actual APs is not detected, and the client only sees all APs of the rogue network. In such a case, the client will not get any warning. The occurrence of such a case should be very rare, but cannot be denied.

Another case when the software will not perform normally is when the client cannot connect to the actual APs even when they are seen. This can be made possible if the attacker floods the real APs with association requests. Any client will then witness a denial of service. In such a case, the proposed software would not work. However, most of the APs are now equipped well to fight the denial-of-service (DoS) attacks, and as such implementing a DoS attack is not an easy task.

The functioning of the software is based on connecting to different APs by identifying their MAC addresses. If however, the MAC address of the APs is spoofed, the software will not be able to differentiate between APs depending on their MAC addresses. The software should then also look at the channel numbers used, and the manufacturer of the AP to differentiate between any two APs with the same MAC address. The existing solutions to fight MAC address spoofing depends on checking the sequence numbers of the frames. If a software is used that is equipped to fight MAC address spoofing, the overall time to perform the check will increase.

This concludes the experimental results of this work, which included implementation of attack, and evaluation of the proposed software.

Chapter 6: Conclusions

The chapter highlights the salient features of this research work and provides a brief summary of the contributions along with the potential for future research.

6.1 Conclusions

The work aims to study an access point spoofing attack on private, open WLANs. One major contribution of this work is to come up with a solution to the rogue AP problem at the client level. The solution's novelty lies in the fact that it does not base the result on traffic monitoring. The existing solutions for finding rogue APs, are mostly very time consuming and difficult to implement. The existing techniques aim at finding out RAPs by analyzing traffic. The need for a client level solution was emphasized. The proposed software provides the clients a way to detect the presence of RAPs in a secured network reducing their dependency on network administrators for such information. The proposed method was found to fulfill the demands of a client side solution. It alerts and warns the user of a possible rogue network in the vicinity before the user actually enters his login id and password on any login page.

The software is easy to implement and use. It is inexpensive. Compared to "wardriving" tools, this software takes much less time.

Experiments were done to find out the probability of the client logging into a rogue network in the absence of such a software, and those results were contrasted with the case when the software was used. In the latter case, the client was alerted of the possible presence of a rogue network such that when he finally logs in to the network, he can be extra cautious of the fake URLs. The client will not know which is the rogue network and which is the actual one, but he will be warned and cautioned. The work also evaluates the performance of the software. The experiments were done to find out the probability of a client discovering and logging into the rogue network in the presence of both rogue and actual APs.

6.2 Future Work

There are several areas of potential future work that could be explored. An important area will be to find a solution in the case the attacker does a MAC address spoofing. Another area would be to work on corner cases. The work also presents a challenge to reduce the overall delay. The high

delay is on account of the delay a client encounters when connecting to, and disconnecting from a WLAN, and starting the browser on the client computer. The code used in this work does everything at the application level. The delay can be reduced by a considerable percentage if the process of connecting to the AP is carried out at driver level, by connecting to the APs in parallel.

Another area of research can be to analyze the geometry of APs in detail. The study attempted to do the experiments based on the assumption that APs were deployed in a straight line in the WLAN. Different topologies of APs can be explored, and experiments can be carried out to find out the best configuration for deploying APs in a network.

Appendix A: Server Set-Up Details

A.1: NAT Router Setting

iptables --flush (Flush all the rules in filter and NAT tables)
iptables --table nat --flush
iptables --delete-chain (Delete all chains that are not in default filter and nat table)
iptables --table nat --delete-chain

Set up IP forwarding ing and Masquerading:
iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE
iptables --append FORWARD --in-interface eth1 -j ACCEPT
echo 1 > /proc/sys/net/ipv4/ip_forward (Enables packet forwarding by kernel) [36]

A.2: DHCP Server Setting

Installation: sudo vi /etc/default/dhcp3-server

Configuration: Make the following changes in /etc/dhcp3/dhcpd.conf file:

default-lease-time 600;
max-lease-time 7200;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name “yourdomainname.com”;

subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.10 192.168.1.200;

} [16,33]

Run: /etc/init.d/dhcp3-server start

A.3: DNS Server Setting

1. Installation:
   apt-get install bind9

2. Configuration:

   i. To add a DNS zone to BIND9, turning BIND9 into a Primary Master server, all you have to do is edit named.conf.local [5]:

```
[...]
    zone "example.com" {
type master;
file "/etc/bind/db.example.com";
};
[...]
```

   ii. Create an **A record** for ns.example.com the name server in this example:

```
;
; BIND data file for local loopback interface
;
    $TTL    604800
    @       IN      SOA     ns.example.com. root.example.com. (
                                ; Serial
        604800              ; Refresh
        86400               ; Retry
        2419200             ; Expire
        604800 )            ; Negative Cache TTL
        ;
    @       IN      NS      ns.example.com.
    @       IN      A       192.168.1.10
    box     IN      A       192.168.1.10
```

    [5], [34]

3. Run: sudo /etc/init.d/bind9 restart

Appendix B: Root Certificate Details

B.1: OpenSSL Steps

1. Create a Certificate Authority:

Create Directories: cd && mkdir -p myCA/signedcerts && mkdir myCA/private && cd myCA
Create DB: cd && mkdir -p myCA/signedcerts && mkdir myCA/private && cd myCA (create DB)
Configuration file: edit the file ~/myCA/caconfig.cnf, and insert the following content into the file:

```
# Default configuration to use when one is not provided on the command line.
#
[ ca ]
default_ca      = local_ca
#
#
# Default location of directories and files needed to generate certificates.
#
[ local_ca ]
dir             = /home/<username>/myCA
certificate     = $dir/cacert.pem
database        = $dir/index.txt
new_certs_dir   = $dir/signedcerts
private_key     = $dir/private/cakey.pem
serial          = $dir/serial
#
#
# Default expiration and encryption policies for certificates.
#
default_crl_days        = 365
default_days            = 1825
default_md              = md5
#
policy          = local_ca_policy
x509_extensions = local_ca_extensions
#
#
# Default policy to use when generating server certificates.  The following
# fields must be defined in the server certificate.
#
[ local_ca_policy ]
commonName              = supplied
```

```
stateOrProvinceName     = supplied
countryName             = supplied
emailAddress            = supplied
organizationName        = supplied
organizationalUnitName  = supplied
#
#
# x509 extensions to use when generating server certificates.
#
[ local_ca_extensions ]
subjectAltName          = DNS:alt.tradeshowhell.com
basicConstraints        = CA:false
nsCertType              = server
#
#
# The default root certificate generation policy.
#
[ req ]
default_bits    = 2048
default_keyfile = /home/<username>/myCA/private/cakey.pem
default_md      = md5
#
prompt                  = no
distinguished_name      = root_ca_distinguished_name
x509_extensions         = root_ca_extensions
#
#
# Root Certificate Authority distinguished name.  Change these fields to match
# your local environment!
#
[ root_ca_distinguished_name ]
commonName              = MyOwn Root Certificate Authority
stateOrProvinceName     = NC
countryName             = US
emailAddress            = root@tradeshowhell.com
organizationName        = Trade Show Hell
organizationalUnitName  = IT Department
#
[ root_ca_extensions ]
basicConstraints        = CA:true
```

2. Generate the Certificate Authority Root Certificate and Key, by issuing a few commands:

```
export OPENSSL_CONF=~/myCA/caconfig.cnf
```

The previous command sets an environment variable, OPENSSL_CONF, which forces the openssl tool to look for a configuration file in an alternative location (in this case, ~/myCA/caconfig.cnf).

Now, generate the CA certificate and key with the following command:

openssl req -x509 -newkey rsa:2048 -out cacert.pem -outform PEM -days 1825

You should be prompted for a passphrase, and see output similar to this:

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/home/username/myCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
```

Server Configuration File:

Create the server configuration file, by editing ~/myCA/exampleserver.cnf with your favorite text editor. Add this example content:

```
#
# exampleserver.cnf
#
[ req ]
prompt                 = no
distinguished_name     = server_distinguished_name
[ server_distinguished_name ]
commonName             = tradeshowhell.com
stateOrProvinceName    = NC
countryName            = US
emailAddress           = root@tradeshowhell.com
organizationName       = My Organization Name
organizationalUnitName  = Subunit of My Large Organization
```

Be sure to change the values under server_distinguished_name especially the commonName value. The commonName value must match the host name, or CNAME for the host you wish to use the key for. If the commonName does not match the intended hostname, then host / certificate mismatch errors will appear in the client applications of clients attempting to access the server.

3. Sign the certificate as follows:

openssl ca -in tempreq.pem -out server_crt.pem

You will be prompted for the passphrase of the CA key as created in the Certificate Authority setup from above. Enter this passphrase at the prompt, and you will then be prompted to confirm the information in the exampleserver.cnf, and finally asked to confirm signing the certificate. Output should be similar to this:

```
Using configuration from /home/username/myCA/caconfig.cnf
Enter pass phrase for /home/username/myCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :PRINTABLE:'tradeshowhell.com'
stateOrProvinceName   :PRINTABLE:'NC'
countryName           :PRINTABLE:'US'
emailAddress          :IA5STRING:'root@tradeshowhell.com'
organizationName      :PRINTABLE:'Trade Show Hell'
organizationalUnitName:PRINTABLE:'Black Ops'
Certificate is to be certified until Jan  4 21:50:08 2011 GMT (1825 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated.

The server certificate and key are:

server_crt.pem : Server application certificate file

server_key.pem : Server application key file [35]
```

Appendix C: List of Abbreviations

AP	Access Point
BSSID	Basic Service Set Identifier
CA	Certificate Authority
CN	Common Name
CRL	Certificate Revocation List
DB	Database
DHCP	Dynamic Host Configuration Protocol
DN	Distinguished Name
DNS	Domain Name System
DoS	Denial Of Service
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IE	Internet Explorer
MAC	Media Access Control
NAT	Network Address Translation
PKI	Public Key Infrastructure
RA	Registration Authority
RAP	Rogue Access Point
RF	Radio Frequency
SSID	Service Set Identifier
SSL	Secure Sockets Layer

Bibliography

1. Adya, A., Bahl, P., Chandra, R., and Qiu, L, "*Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks",* In MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking (New York, NY, USA, 2004), ACM Press, pages 30-44
2. "AirDefense Warning Against Phishing", http://www.thefreelibrary.com/AirDefense+Warns+Corporate+Executives+of+New+Phishing+Scam+Targeting...-a0132234833
3. "AirMagnet", http://www.airmagent.com
4. "Anti Phishing Working Group, Phishing Activity Trends Report, 1st Half, 2009", http://www.antiphishing.org/reports/apwg_report_h1_2009.pdf
5. "Bind9ServerHowto", https://help.ubuntu.com/community/BIND9ServerHowto
6. "CBSNews.com", http://www.cbsnews.com/stories/2009/10/06/scitech/pcanswer/main5367811.shtml
7. "Calling ID Toolbar", http://www.callingid.com/DesktopSolutions/CallingIDToolbar.aspx
8. "Cisco Wireless LAN Solution Engine", http://www.cisco.com/en/US/products/sw/cscowork/ps3915/index.html
9. "CNetNews", http://news.cnet.com/8301-19518_3-10368801-238.html
10. "Cryptocard", http://www.cryptocard.com/company/
11. "CryptoCards", https://www.racf.bnl.gov/docs/authentication/cryptocards
12. Dhamija, J. Tygar, and M. Hearst, "*Why Phishing Works*" In Human Factors in Computing Systems (CHI 2006), Quebec, Canada, Apr. 22–27, 2006.
13. Dhamija, R. & J. D. Tygar, " *The Battle Against Phishing: Dynamic Security Skins*" *Proc. SOUPS* (2005).
14. E. Kirda and C. Kruegel, "*Protecting users against phishing attacks with antiphish*", In Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC), pages 517–524, 2005.
15. "Facebook Phishing Article", http://www.cnn.com/2009/TECH/04/30/facebook.phishing.attacks/index.html

16. "How to Install and Configure DHCP Server in Ubuntu Server", http://www.ubuntugeek.com/how-to-install-and-configure-dhcp-server-in-ubuntu-server.html
17. I. Martinovic, F. A. Zdarsky, A. Bachorek, C. Jung, and J. B. Schmitt, "*Phishing in the Wireless: Implementation and Analysis*", In Proceedings of the 22nd IFIP International Information Security Conference (SEC 2007), pages 145–156, May 2007.
18. "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, http://www.ietf.org/rfc/rfc3280.txt
19. "Introduction to Public-Key Cryptography", http://docs.sun.com/source/816-6154-10/contents.htm
20. "Introduction to SSL", http://docs.sun.com/source/816-6156-10/contents.htm
21. "IRS: Suspicious email/Phishing", http://www.irs.gov/newsroom/article/0,,id=155682,00.html
22. Krammer, V, "*Phishing Defense against IDN Address Spoofing Attacks",* In: Proceedings of the 4th Annual Privacy Security Trust Conference 2006 (PST 2006), October 2006, pp. 275–284. ACM Press, New York (2006)
23. L. Ma, A. Y. Teymorian, X. Cheng, and M. Song, "*RAP: Protecting commodity wi-fi networks from rogue access points*," in QShine '07:Proceedings of the 4th international conference on Quality of service in heterogeneous wired/wireless networks, 2007
24. Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, John C. Mitchell, *Client Side Defense Against Web-based Identity Theft*, http://crypto.stanford.edu/SpoofGuard/webspoof.pdf
25. P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "*Enhancing the security of corporate wi-fi networks using DAIR",* In *MobiSys*, 2006.
26. Paul Knickerbocker, "Combating Phishing Through Zero-Knowledge Authentication", Thesis, Department of Computer and Information Science, Master of Science, June 2008
27. S. Shetty, M. Song, and L. Ma, "*Rogue access point detection by analyzing network traffic characteristics*," in MILCOM, Orlando, Florida, October 2007

28. “SANS Institute InfoSec Reading Room”, http://www.sans.org/reading_room/whitepapers/threats/ssl_maninthemiddle_attacks_480?show=480.php&cat=threats
29. “Security and Cryptography”, http://forums.devshed.com/security-and-cryptography-17/ssl-man-in-the-middle-attack-86557.html
30. “ShortNews.com”, http://www.shortnews.com/start.cfm?id=81015
31. “SSL's Credibility as Phishing Defense Is Tested”, http://news.netcraft.com/archives/2004/03/08/ssls_credibility_as_phishing_defense_is_tested.html
32. “Twitter Blog”, http://blog.twitter.com/2009/01/gone-phishing.html
33. “Ubuntu Documentation, dhcp3-server”, https://help.ubuntu.com/community/dhcp3-server
34. “Ubuntu Documentation, DNS Configuration”, https://help.ubuntu.com/8.04/serverguide/C/dns-configuration.html
35. “Ubuntu Documentation, OpenSSL”, https://help.ubuntu.com/community/OpenSSL#About%20OpenSSL
36. “Using Linux iptables or ipchains to set up an internet gateway/ firewall/ router for home or office”, http://www.yolinux.com/TUTORIALS/LinuxTutorialIptablesNetworkGateway.html
37. “Verisign FAQs, SSL Basics”, http://www.verisign.com/ssl/ssl-information-center/ssl-basics/index.html#a7
38. “What are Digital Certificates?”, http://www.tech-faq.com/digital-certificates.shtml
39. “Wimetrics”, http://www.wimetrics.com/
40. “WOT”, http://www.mywot.com/
41. Y. F. Anthony, W. Liu, X. Deng. “*Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)”,* IEEE Transactions on Dependable and Secure Computing, October 2006, Volume 3 (4), 301-311